



Welcome to [E-XFL.COM](https://www.e-xfl.com)

### What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

### Applications of "[Embedded - Microcontrollers](#)"

#### Details

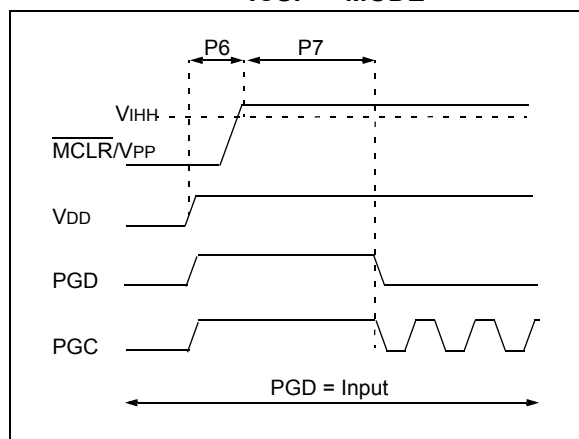
Product Status	Obsolete
Core Processor	dsPIC
Core Size	16-Bit
Speed	20 MIPS
Connectivity	CANbus, I <sup>2</sup> C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	52
Program Memory Size	132KB (44K x 24)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	6K x 8
Voltage - Supply (Vcc/Vdd)	2.5V ~ 5.5V
Data Converters	A/D 16x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/dspic30f6011at-20e-pt">https://www.e-xfl.com/product-detail/microchip-technology/dspic30f6011at-20e-pt</a>

# dsPIC30F Flash Programming Specification

## 5.2 Entering Enhanced ICSP Mode

The Enhanced ICSP mode is entered by holding PGC and PGD high, and then raising MCLR/VPP to VIH (high voltage), as illustrated in Figure 5-2. In this mode, the code memory, data EEPROM and Configuration bits can be efficiently programmed using the programming executive commands that are serially transferred using PGC and PGD.

**FIGURE 5-2: ENTERING ENHANCED ICSP™ MODE**



**Note 1:** The sequence that places the device into Enhanced ICSP mode places all unused I/Os in the high-impedance state.

**2:** Before entering Enhanced ICSP mode, clock switching must be disabled using ICSP, by programming the FCKSM<1:0> bits in the FOSC Configuration register to '11' or '10'.

**3:** When in Enhanced ICSP mode, the SPI output pin (SDO1) will toggle while the device is being programmed.

## 5.3 Chip Erase

Before a chip can be programmed, it must be erased. The Bulk Erase command (**ERASEB**) is used to perform this task. Executing this command with the MS command field set to 0x3 erases all code memory, data EEPROM and code-protect Configuration bits. The Chip Erase process sets all bits in these three memory regions to '1'.

Since non-code-protect Configuration bits cannot be erased, they must be manually set to '1' using multiple **PROGC** commands. One **PROGC** command must be sent for each Configuration register (see Section 5.7 "Configuration Bits Programming").

If Advanced Security features are enabled, then individual Segment Erase operations would need to be performed, depending on which segment needs to be programmed at a given stage of system programming. The user should have the flexibility to select specific segments for programming.

**Note:** The Device ID registers cannot be erased. These registers remain intact after a Chip Erase is performed.

## 5.4 Blank Check

The term "Blank Check" means to verify that the device has been successfully erased and has no programmed memory cells. A blank or erased memory cell reads as '1'. The following memories must be blank checked:

- All implemented code memory
- All implemented data EEPROM
- All Configuration bits (for their default value)

The Device ID registers (0xFF0000:0xFF0002) can be ignored by the Blank Check since this region stores device information that cannot be erased. Additionally, all unimplemented memory space should be ignored from the Blank Check.

The **QBLANK** command is used for the Blank Check. It determines if the code memory and data EEPROM are erased by testing these memory regions. A 'BLANK' or 'NOT BLANK' response is returned. The **READD** command is used to read the Configuration registers. If it is determined that the device is not blank, it must be erased (see Section 5.3 "Chip Erase") before attempting to program the chip.

# dsPIC30F Flash Programming Specification

## 5.5.3 PROGRAMMING VERIFICATION

Once code memory is programmed, the contents of memory can be verified to ensure that programming was successful. Verification requires code memory to be read back and compared against the copy held in the programmer's buffer.

The `READP` command can be used to read back all the programmed code memory.

Alternatively, you can have the programmer perform the verification once the entire device is programmed using a checksum computation, as described in [Section 6.8 "Checksum Computation"](#).

## 5.6 Data EEPROM Programming

### 5.6.1 OVERVIEW

The panel architecture for the data EEPROM memory array consists of 128 rows of sixteen 16-bit data words. Each panel stores 2K words. All devices have either one or no memory panels. Devices with data EEPROM provide either 512 words, 1024 words or 2048 words of memory on the one panel (see [Table 5-3](#)).

**TABLE 5-3: DATA EEPROM SIZE**

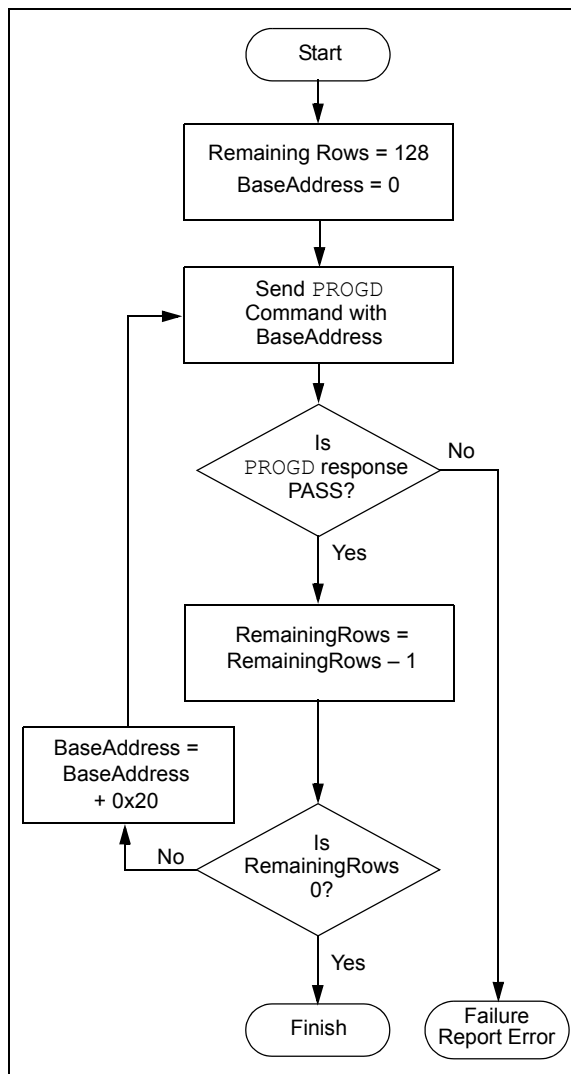
Device	Data EEPROM Size (Words)	Number of Rows
dsPIC30F2010	512	32
dsPIC30F2011	0	0
dsPIC30F2012	0	0
dsPIC30F3010	512	32
dsPIC30F3011	512	32
dsPIC30F3012	512	32
dsPIC30F3013	512	32
dsPIC30F3014	512	32
dsPIC30F4011	512	32
dsPIC30F4012	512	32
dsPIC30F4013	512	32
dsPIC30F5011	512	32
dsPIC30F5013	512	32
dsPIC30F5015	512	32
dsPIC30F5016	512	32
dsPIC30F6010	2048	128
dsPIC30F6010A	2048	128
dsPIC30F6011	1024	64
dsPIC30F6011A	1024	64
dsPIC30F6012	2048	128
dsPIC30F6012A	2048	128
dsPIC30F6013	1024	64
dsPIC30F6013A	1024	64
dsPIC30F6014	2048	128
dsPIC30F6014A	2048	128
dsPIC30F6015	2048	128

## 5.6.2 PROGRAMMING METHODOLOGY

The programming executive uses the `PROGD` command to program the data EEPROM. [Figure 5-4](#) illustrates the flowchart of the process. Firstly, the number of rows to program (RemainingRows) is based on the device size, and the destination address (BaseAddress) is set to '0'. In this example, 128 rows (2048 words) of data EEPROM will be programmed.

The first `PROGD` command programs the first row of data EEPROM. Once the command completes successfully, 'RemainingRows' is decremented by 1 and compared with 0. Since there are 127 more rows to program, 'BaseAddress' is incremented by 0x20 to point to the next row of data EEPROM. This process is then repeated until all 128 rows of data EEPROM are programmed.

**FIGURE 5-4: FLOWCHART FOR PROGRAMMING dsPIC30F6014A DATA EEPROM**



# dsPIC30F Flash Programming Specification

## 5.6.3 PROGRAMMING VERIFICATION

Once the data EEPROM is programmed, the contents of memory can be verified to ensure that the programming was successful. Verification requires the data EEPROM to be read back and compared against the copy held in the programmer's buffer. The `READD` command reads back the programmed data EEPROM.

Alternatively, the programmer can perform the verification once the entire device is programmed using a checksum computation, as described in [Section 6.8 "Checksum Computation"](#).

**Note:** `TBLRD` instructions executed within a `REPEAT` loop must not be used to read from Data EEPROM. Instead, it is recommended to use PSV access.

## 5.7 Configuration Bits Programming

### 5.7.1 OVERVIEW

The dsPIC30F has Configuration bits stored in seven 16-bit registers. These bits can be set or cleared to select various device configurations. There are two types of Configuration bits: system-operation bits and code-protect bits. The system-operation bits determine the power-on settings for system-level components such as the oscillator and Watchdog Timer. The code-protect bits prevent program memory from being read and written.

The FOSC Configuration register has three different register descriptions, based on the device. The FOSC Configuration register description for the dsPIC30F2010 and dsPIC30F6010/6011/6012/6013/6014 devices are shown in [Table 5-4](#).

**Note:** If user software performs an erase operation on the configuration fuse, it must be followed by a write operation to this fuse with the desired value, even if the desired value is the same as the state of the erased fuse.

The FOSC Configuration register description for the dsPIC30F4011/4012 and dsPIC30F5011/5013 devices is shown in [Table 5-5](#).

The FOSC Configuration register description for all remaining devices (dsPIC30F2011/2012, dsPIC30F3010/3011/3012/3013, dsPIC30F3014/4013, dsPIC30F5015 and dsPIC30F6011A/6012A/6013A/6014A) is shown in [Table 5-6](#). Always use the correct register descriptions for your target processor.

The `FWDT`, `FBORPOR`, `FBS`, `FSS`, `FGS` and `FICD` Configuration registers are not device-dependent. The register descriptions for these Configuration registers are shown in [Table 5-7](#).

The Device Configuration register maps are shown in [Table 5-8](#) through [Table 5-11](#).

**TABLE 5-4: FOSC CONFIGURATION BITS DESCRIPTION FOR dsPIC30F2010 AND dsPIC30F6010/6011/6012/6013/6014**

Bit Field	Register	Description
FCKSM<1:0>	FOSC	<b>Clock Switching Mode</b> 1x = Clock switching is disabled, Fail-Safe Clock Monitor is disabled 01 = Clock switching is enabled, Fail-Safe Clock Monitor is disabled 00 = Clock switching is enabled, Fail-Safe Clock Monitor is enabled
FOS<1:0>	FOSC	<b>Oscillator Source Selection on POR</b> 11 = Primary Oscillator 10 = Internal Low-Power RC Oscillator 01 = Internal Fast RC Oscillator 00 = Low-Power 32 kHz Oscillator (Timer1 Oscillator)
FPR<3:0>	FOSC	<b>Primary Oscillator Mode</b> 1111 = ECIO w/PLL 16X – External Clock mode with 16X PLL. OSC2 pin is I/O 1110 = ECIO w/PLL 8X – External Clock mode with 8X PLL. OSC2 pin is I/O 1101 = ECIO w/PLL 4X – External Clock mode with 4X PLL. OSC2 pin is I/O 1100 = ECIO – External Clock mode. OSC2 pin is I/O 1011 = EC – External Clock mode. OSC2 pin is system clock output (Fosc/4) 1010 = Reserved (do not use) 1001 = ERC – External RC Oscillator mode. OSC2 pin is system clock output (Fosc/4) 1000 = ERCIO – External RC Oscillator mode. OSC2 pin is I/O 0111 = XT w/PLL 16X – XT Crystal Oscillator mode with 16X PLL 0110 = XT w/PLL 8X – XT Crystal Oscillator mode with 8X PLL 0101 = XT w/PLL 4X – XT Crystal Oscillator mode with 4X PLL 0100 = XT – XT Crystal Oscillator mode (4 MHz-10 MHz crystal) 001x = HS – HS Crystal Oscillator mode (10 MHz-25 MHz crystal) 000x = XTL – XTL Crystal Oscillator mode (200 kHz-4 MHz crystal)

# dsPIC30F Flash Programming Specification

**TABLE 5-6: FOSC CONFIGURATION BITS DESCRIPTION FOR dsPIC30F2011/2012, dsPIC30F3010/3011/3012/3013/3014, dsPIC30F4013, dsPIC30F5015/5016, dsPIC30F6010A/6011A/6012A/6013A/6014A AND dsPIC30F6015 (CONTINUED)**

Bit Field	Register	Description
FPR<4:0>	FOSC	<b>Alternate Oscillator Mode (when FOS&lt;2:0&gt; = 011b)</b> 1xxxx = Reserved (do not use) 0111x = Reserved (do not use) 01101 = Reserved (do not use) 01100 = ECIO – External clock. OSC2 pin is I/O 01011 = EC – External clock. OSC2 pin is system clock output (Fosc/4) 01010 = Reserved (do not use) 01001 = ERC – External RC oscillator. OSC2 pin is system clock output (Fosc/4) 01000 = ERCIO – External RC oscillator. OSC2 pin is I/O 00111 = Reserved (do not use) 00110 = Reserved (do not use) 00101 = Reserved (do not use) 00100 = XT – XT crystal oscillator (4 MHz-10 MHz crystal) 00010 = HS – HS crystal oscillator (10 MHz-25 MHz crystal) 00001 = Reserved (do not use) 00000 = XTL – XTL crystal oscillator (200 kHz-4 MHz crystal)

# dsPIC30F Flash Programming Specification

## 8.5.5 PROGP COMMAND

15	12	11	8	7	0
Opcode		Length			
Reserved			Addr_MSB		
Addr_LS					
D_1					
D_2					
...					
D_N					

Field	Description
Opcode	0x5
Length	0x33
Reserved	0x0
Addr_MSB	MSB of 24-bit destination address
Addr_LS	LS 16 bits of 24-bit destination address
D_1	16-bit data word 1
D_2	16-bit data word 2
...	16-bit data word 3 through 47
D_48	16-bit data word 48

The **PROGP** command instructs the programming executive to program one row of code memory (32 instruction words) to the specified memory address. Programming begins with the row address specified in the command. The destination address should be a multiple of 0x40.

The data to program to memory, located in command words D\_1 through D\_48, must be arranged using the packed instruction word format shown in [Figure 8-2](#).

After all data has been programmed to code memory, the programming executive verifies the programmed data against the data in the command.

### Expected Response (2 words):

0x1500  
0x0002

**Note:** Refer to [Table 5-2](#) for code memory size information.

## 8.5.6 PROGC COMMAND

15	12	11	8	7	0
Opcode		Length			
Reserved			Addr_MSB		
Addr_LS					
Data					

Field	Description
Opcode	0x6
Length	0x4
Reserved	0x0
Addr_MSB	MSB of 24-bit destination address
Addr_LS	LS 16 bits of 24-bit destination address
Data	Data to program

The **PROGC** command programs data to the specified Configuration register and verifies the programming. Configuration registers are 16 bits wide, and this command allows one Configuration register to be programmed.

### Expected Response (2 words):

0x1600  
0x0002

**Note:** This command can only be used for programming Configuration registers.

# dsPIC30F Flash Programming Specification

## 8.5.7 ERASEB COMMAND

15	12	11		2	0
Opcode		Length			
Reserved					MS

Field	Description
Opcode	0x7
Length	0x2
Reserved	0x0
MS	Select memory to erase: 0x0 = All Code in General Segment 0x1 = All Data EEPROM in General Segment 0x2 = All Code and Data EEPROM in General Segment, interrupt vectors and FGS Configuration register 0x3 = Full Chip Erase 0x4 = All Code and Data EEPROM in Boot, Secure and General Segments, and FBS, FSS and FGS Configuration registers 0x5 = All Code and Data EEPROM in Secure and General Segments, and FSS and FGS Configuration registers 0x6 = All Data EEPROM in Boot Segment 0x7 = All Data EEPROM in Secure Segment

The **ERASEB** command performs a Bulk Erase. The MS field selects the memory to be bulk erased, with options for erasing Code and/or Data EEPROM in individual memory segments.

When Full Chip Erase is selected, the following memory regions are erased:

- All code memory (even if code-protected)
- All data EEPROM
- All code-protect Configuration registers

Only the executive code memory, Unit ID, device ID and Configuration registers that are not code-protected remain intact after a Chip Erase.

### Expected Response (2 words):

0x1700  
0x0002

**Note:** A Full Chip Erase cannot be performed in low-voltage programming systems ( $V_{DD}$  less than 4.5 volts). **ERASED** and **ERASEP** must be used to erase code memory, executive memory and data memory. Alternatively, individual Segment Erase operations may be performed.

## 8.5.8 ERASED COMMAND

15	12	11		8	7		0
Opcode		Length					
Num_Rows				Addr_MSB			
Addr_LS							

Field	Description
Opcode	0x8
Length	0x3
Num_Rows	Number of rows to erase (max of 128)
Addr_MSB	MSB of 24-bit base address
Addr_LS	LS 16 bits of 24-bit base address

The **ERASED** command erases the specified number of rows of data EEPROM from the specified base address. The specified base address must be a multiple of 0x20. Since the data EEPROM is mapped to program space, a 24-bit base address must be specified.

After the erase is performed, all targeted bytes of data EEPROM will contain 0xFF.

### Expected Response (2 words):

0x1800  
0x0002

**Note:** The **ERASED** command cannot be used to erase the Configuration registers or device ID. Code-protect Configuration registers can only be erased with the **ERASEB** command, while the device ID is read-only.

# dsPIC30F Flash Programming Specification

## 11.0 ICSP™ MODE

### 11.1 ICSP Mode

ICSP mode is a special programming protocol that allows you to read and write to the dsPIC30F programming executive. The ICSP mode is the second (and slower) method used to program the device. This mode also has the ability to read the contents of executive memory to determine whether the programming executive is present. This capability is accomplished by applying control codes and instructions serially to the device using pins PGC and PGD.

In ICSP mode, the system clock is taken from the PGC pin, regardless of the device's oscillator Configuration bits. All instructions are first shifted serially into an internal buffer, then loaded into the Instruction register and executed. No program fetching occurs from internal memory. Instructions are fed in 24 bits at a time. PGD is used to shift data in and PGC is used as both the serial shift clock and the CPU execution clock.

Data is transmitted on the rising edge and latched on the falling edge of PGC. For all data transmissions, the Least Significant bit (LSb) is transmitted first.

**Note 1:** During ICSP operation, the operating frequency of PGC must not exceed 5 MHz.

**2:** Because ICSP is slower, it is recommended that only Enhanced ICSP (E-ICSP) mode be used for device programming, as described in [Section 5.1 "Overview of the Programming Process"](#).

### 11.2 ICSP Operation

Upon entry into ICSP mode, the CPU is idle. Execution of the CPU is governed by an internal state machine. A 4-bit control code is clocked in using PGC and PGD, and this control code is used to command the CPU (see [Table 11-1](#)).

The SIX control code is used to send instructions to the CPU for execution, while the REGOUT control code is used to read data out of the device via the VISI register. The operation details of ICSP mode are provided in [Section 11.2.1 "SIX Serial Instruction Execution"](#) and [Section 11.2.2 "REGOUT Serial Instruction Execution"](#).

**TABLE 11-1: CPU CONTROL CODES IN ICSP™ MODE**

4-bit Control Code	Mnemonic	Description
0000b	SIX	Shift in 24-bit instruction and execute.
0001b	REGOUT	Shift out the VISI register.
0010b-1111b	N/A	Reserved.

#### 11.2.1 SIX SERIAL INSTRUCTION EXECUTION

The SIX control code allows execution of dsPIC30F assembly instructions. When the SIX code is received, the CPU is suspended for 24 clock cycles as the instruction is then clocked into the internal buffer. Once the instruction is shifted in, the state machine allows it to be executed over the next four clock cycles. While the received instruction is executed, the state machine simultaneously shifts in the next 4-bit command (see [Figure 11-2](#)).

**Note 1:** Coming out of the ICSP entry sequence, the first 4-bit control code is always forced to SIX and a forced NOP instruction is executed by the CPU. Five additional PGC clocks are needed on start-up, thereby resulting in a 9-bit SIX command instead of the normal 4-bit SIX command. After the forced SIX is clocked in, ICSP operation resumes as normal (the next 24 clock cycles load the first instruction word to the CPU). See [Figure 11-1](#) for details.

**2:** TBLRDH, TBLRDL, TBLWTH and TBLWTL instructions must be followed by a NOP instruction.



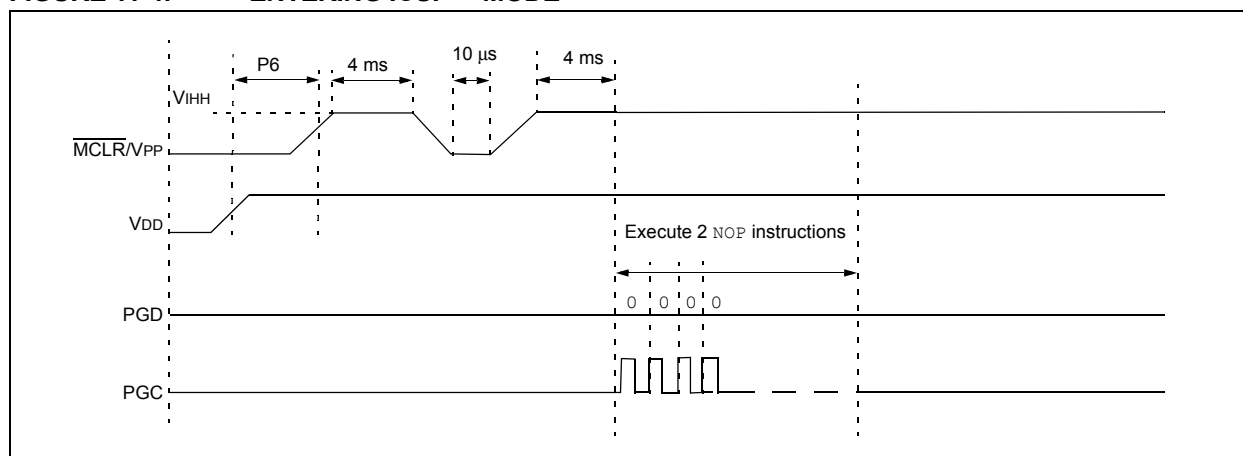
# dsPIC30F Flash Programming Specification

## 11.3 Entering ICSP Mode

The ICSP mode is entered by holding PGC and PGD low, raising MCLR/VPP to VIH (high voltage), and then performing additional steps as illustrated in [Figure 11-4](#).

- Note 1:** The sequence that places the device into ICSP mode places all unused I/O pins to the high-impedance state.
- 2:** Once ICSP mode is entered, the PC is set to 0x0 (the Reset vector).
- 3:** Before leaving the Reset vector, execute two GOTO instructions, followed by a single NOP instruction must be executed.

**FIGURE 11-4: ENTERING ICSP™ MODE**



# dsPIC30F Flash Programming Specification

Table 11-4 shows the ICSP programming process for bulk-erasing program memory. This process includes the ICSP command code, which must be transmitted (for each instruction) to the Least Significant bit first using the PGC and PGD pins (see Figure 11-2).

If an individual Segment Erase operation is required, the NVMCON value must be replaced by the value for the corresponding Segment Erase operation.

**Note:** Program memory must be erased before writing any data to program memory.

**TABLE 11-4: SERIAL INSTRUCTION EXECUTION FOR BULK ERASING PROGRAM MEMORY (ONLY IN NORMAL-VOLTAGE SYSTEMS)**

Command (Binary)	Data (Hexadecimal)	Description
<b>Step 1: Exit the Reset vector.</b>		
0000	040100	GOTO 0x100
0000	040100	GOTO 0x100
0000	000000	NOP
<b>Step 2: Set NVMCON to program the FBS Configuration register.<sup>(1)</sup></b>		
0000	24008A	MOV #0x4008, W10
0000	883B0A	MOV W10, NVMCON
<b>Step 3: Initialize the TBLPAG and write pointer (W7) for TBLWT instruction for Configuration register.<sup>(1)</sup></b>		
0000	200F80	MOV #0xF8, W0
0000	880190	MOV W0, TBLPAG
0000	200067	MOV #0x6, W7
<b>Step 4: Load the Configuration Register data to W6.<sup>(1)</sup></b>		
0000	EB0300	CLR W6
0000	000000	NOP
<b>Step 5: Load the Configuration Register write latch. Advance W7 to point to next Configuration register.<sup>(1)</sup></b>		
0000	BB1B86	TBLWTL W6, [W7++]
<b>Step 6: Unlock the NVMCON for programming the Configuration register.<sup>(1)</sup></b>		
0000	200558	MOV #0x55, W8
0000	200AA9	MOV #0xAA, W9
0000	883B38	MOV W8, NVMKEY
0000	883B39	MOV W9, NVMKEY
<b>Step 7: Initiate the programming cycle.<sup>(1)</sup></b>		
0000	A8E761	BSET NVMCON, #WR
0000	000000	NOP
0000	000000	NOP
—	—	Externally time 2 ms
0000	000000	NOP
0000	000000	NOP
0000	A9E761	BCLR NVMCON, #WR
0000	000000	NOP
0000	000000	NOP
<b>Step 8: Repeat steps 5-7 one time to program 0x0000 to RESERVED2 Configuration register.<sup>(1)</sup></b>		
<b>Step 9: Set the NVMCON to erase all Program Memory.</b>		
00000	2407FA	MOV #0x407F, W10
0000	883B0A	MOV W10, NVMCON
<b>Step 10: Unlock the NVMCON for programming.</b>		

**Note 1:** Steps 2-8 are only required for the dsPIC30F5011/5013 devices. These steps may be skipped for all other devices in the dsPIC30F family.

# dsPIC30F Flash Programming Specification

## 11.6 Erasing Program Memory in Low-Voltage Systems

The procedure for erasing program memory (all code memory and data memory) in low-voltage systems (with VDD between 2.5 volts and 4.5 volts) is quite different than the procedure for erasing program memory in normal-voltage systems. Instead of using a Bulk Erase operation, each region of memory must be individually erased by row. Namely, all of the code memory, executive memory and data memory must be erased one row at a time. This procedure is detailed in [Table 11-5](#).

Due to security restrictions, the FBS, FSS and FGS register cannot be erased in low-voltage systems. Once any bits in the FGS register are programmed to '0', they can only be set back to '1' by performing a Bulk Erase in a normal-voltage system. Alternatively, a Segment Erase operation can be performed instead of a Bulk Erase.

Normal-voltage systems can also be used to erase program memory as shown in [Table 11-5](#). However, since this method is more time-consuming and does not clear the code-protect bits, it is not recommended.

**Note:** Program memory must be erased before writing any data to program memory.

**TABLE 11-5: SERIAL INSTRUCTION EXECUTION FOR ERASING PROGRAM MEMORY (EITHER IN LOW-VOLTAGE OR NORMAL-VOLTAGE SYSTEMS)**

Command (Binary)	Data (Hexadecimal)	Description
<b>Step 1: Exit the Reset vector.</b>		
0000	040100	GOTO 0x100
0000	040100	GOTO 0x100
0000	000000	NOP
<b>Step 2: Initialize NVMADR and NVMADRU to erase code memory and initialize W7 for row address updates.</b>		
0000	EB0300	CLR W6
0000	883B16	MOV W6, NVMADR
0000	883B26	MOV W6, NVMADRU
0000	200407	MOV #0x40, W7
<b>Step 3: Set NVMCON to erase 1 row of code memory.</b>		
0000	24071A	MOV #0x4071, W10
0000	883B0A	MOV W10, NVMCON
<b>Step 4: Unlock the NVMCON to erase 1 row of code memory.</b>		
0000	200558	MOV #0x55, W8
0000	883B38	MOV W8, NVMKEY
0000	200AA9	MOV #0xAA, W9
0000	883B39	MOV W9, NVMKEY
<b>Step 5: Initiate the erase cycle.</b>		
0000	A8E761	BSET NVMCON, #WR
0000	000000	NOP
0000	000000	NOP
—	—	Externally time 'P13a' ms (see <a href="#">Section 13.0 “AC/DC Characteristics and Timing Requirements”</a> )
0000	000000	NOP
0000	000000	NOP
0000	A9E761	BCLR NVMCON, #WR
0000	000000	NOP
0000	000000	NOP

# dsPIC30F Flash Programming Specification

**TABLE 11-5: SERIAL INSTRUCTION EXECUTION FOR ERASING PROGRAM MEMORY  
(EITHER IN LOW-VOLTAGE OR NORMAL-VOLTAGE SYSTEMS) (CONTINUED)**

Command (Binary)	Data (Hexadecimal)	Description
<b>Step 18:</b> Unlock the NVMCON to erase 1 row of data memory.		
0000	200558	MOV #0x55, W8
0000	883B38	MOV W8, NVMKEY
0000	200AA9	MOV #0xAA, W9
0000	883B39	MOV W9, NVMKEY
<b>Step 19:</b> Initiate the erase cycle.		
0000	A8E761	BSET NVMCON, #WR
0000	000000	NOP
0000	000000	NOP
—	—	Externally time 'P13a' ms (see <a href="#">Section 13.0 “AC/DC Characteristics and Timing Requirements”</a> )
0000	000000	NOP
0000	000000	NOP
0000	A9E761	BCLR NVMCON, #WR
0000	000000	NOP
0000	000000	NOP
<b>Step 20:</b> Update the row address stored in NVMADR.		
0000	430307	ADD W6, W7, W6
0000	883B16	MOV W6, NVMADR
<b>Step 21:</b> Reset device internal PC.		
0000	040100	GOTO 0x100
0000	000000	NOP
<b>Step 22:</b> Repeat Steps 17-21 until all rows of data memory are erased.		

# dsPIC30F Flash Programming Specification

**TABLE 11-7: SERIAL INSTRUCTION EXECUTION FOR WRITING CONFIGURATION REGISTERS (CONTINUED)**

Command (Binary)	Data (Hexadecimal)	Description
<b>Step 6:</b> Write the Configuration register data to the write latch and increment the write pointer.		
0000	BB1B96	TBLWTL W6, [W7++]
0000	000000	NOP
0000	000000	NOP
<b>Step 7:</b> Unlock the NVMCON for programming.		
0000	200558	MOV #0x55, W8
0000	883B38	MOV W8, NVMKEY
0000	200AA9	MOV #0xAA, W9
0000	883B39	MOV W9, NVMKEY
<b>Step 8:</b> Initiate the write cycle.		
0000	A8E761	BSET NVMCON, #WR
0000	000000	NOP
0000	000000	NOP
—	—	Externally time 'P12a' ms (see <a href="#">Section 13.0 “AC/DC Characteristics and Timing Requirements”</a> )
0000	000000	NOP
0000	000000	NOP
0000	A9E761	BCLR NVMCON, #WR
0000	000000	NOP
0000	000000	NOP
<b>Step 9:</b> Reset device internal PC.		
0000	040100	GOTO 0x100
0000	000000	NOP
<b>Step 10:</b> Repeat steps 3-9 until all 7 Configuration registers are cleared.		

# dsPIC30F Flash Programming Specification

**TABLE 11-8: SERIAL INSTRUCTION EXECUTION FOR WRITING CODE MEMORY (CONTINUED)**

Command (Binary)	Data (Hexadecimal)	Description
<b>Step 5:</b> Set the read pointer (W6) and load the (next set of) write latches.		
0000	EB0300	CLR W6
0000	000000	NOP
0000	BB0BB6	TBLWTL [W6++], [W7]
0000	000000	NOP
0000	000000	NOP
0000	BBDBB6	TBLWTH.B [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BEBBB6	TBLWTH.B [W6++], [++W7]
0000	000000	NOP
0000	000000	NOP
0000	BB1BB6	TBLWTL [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BB0BB6	TBLWTL [W6++], [W7]
0000	000000	NOP
0000	000000	NOP
0000	BBDBB6	TBLWTH.B [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BEBBB6	TBLWTH.B [W6++], [++W7]
0000	000000	NOP
0000	000000	NOP
0000	BB1BB6	TBLWTL [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
<b>Step 6:</b> Repeat steps 4-5 eight times to load the write latches for 32 instructions.		
<b>Step 7:</b> Unlock the NVMCON for writing.		
0000	200558	MOV #0x55, W8
0000	883B38	MOV W8, NVMKEY
0000	200AA9	MOV #0xAA, W9
0000	883B39	MOV W9, NVMKEY
<b>Step 8:</b> Initiate the write cycle.		
0000	A8E761	BSET NVMCON, #WR
0000	000000	NOP
0000	000000	NOP
—	—	Externally time 'P12a' ms (see <a href="#">Section 13.0 “AC/DC Characteristics and Timing Requirements”</a> )
0000	000000	NOP
0000	000000	NOP
0000	A9E761	BCLR NVMCON, #WR
0000	000000	NOP
0000	000000	NOP
<b>Step 9:</b> Reset device internal PC.		
0000	040100	GOTO 0x100
0000	000000	NOP
<b>Step 10:</b> Repeat steps 2-9 until all code memory is programmed.		

# dsPIC30F Flash Programming Specification

## 11.9 Writing Data EEPROM

The procedure for writing data EEPROM is very similar to the procedure for writing code memory, except that fewer words are programmed in each operation. When writing data EEPROM, one row of data EEPROM is programmed at a time. Each row consists of sixteen 16-bit data words. Since fewer words are programmed

during each operation, only working registers W0:W3 are used as temporary holding registers for the data to be programmed.

Table 11-9 shows the ICSP programming details for writing data EEPROM. Note that a different NVMCON value is required to write to data EEPROM, and that the TBLPAG register is hard-coded to 0x7F (the upper byte address of all locations of data EEPROM).

**TABLE 11-9: SERIAL INSTRUCTION EXECUTION FOR WRITING DATA EEPROM**

Command (Binary)	Data (Hexadecimal)	Description
<b>Step 1: Exit the Reset vector.</b>		
0000	040100	GOTO 0x100
0000	040100	GOTO 0x100
0000	000000	NOP
<b>Step 2: Set the NVMCON to write 16 data words.</b>		
0000	24005A	MOV #0x4005, W10
0000	883B0A	MOV W10, NVMCON
<b>Step 3: Initialize the write pointer (W7) for TBLWT instruction.</b>		
0000	2007F0	MOV #0x7F, W0
0000	880190	MOV W0, TBLPAG
0000	2xxxx7	MOV #<DestinationAddress15:0>, W7
<b>Step 4: Load W0:W3 with the next 4 data words to program.</b>		
0000	2xxxx0	MOV #<WORD0>, W0
0000	2xxxx1	MOV #<WORD1>, W1
0000	2xxxx2	MOV #<WORD2>, W2
0000	2xxxx3	MOV #<WORD3>, W3
<b>Step 5: Set the read pointer (W6) and load the (next set of) write latches.</b>		
0000	EB0300	CLR W6
0000	000000	NOP
0000	BB1BB6	TBLWTL [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BB1BB6	TBLWTL [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BB1BB6	TBLWTL [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BB1BB6	TBLWTL [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
<b>Step 6: Repeat steps 4-5 four times to load the write latches for 16 data words.</b>		

# dsPIC30F Flash Programming Specification

## 11.12 Reading Data Memory

The procedure for reading data memory is similar to that of reading code memory, except that 16-bit data words are read instead of 24-bit words. Since less data is read in each operation, only working registers W0:W3 are used as temporary holding registers for the data to be read.

Table 11-12 shows the ICSP programming details for reading data memory. Note that the TBLPAG register is hard-coded to 0x7F (the upper byte address of all locations of data memory).

**TABLE 11-12: SERIAL INSTRUCTION EXECUTION FOR READING DATA MEMORY**

Command (Binary)	Data (Hexadecimal)	Description
<b>Step 1: Exit the Reset vector.</b>		
0000	040100	GOTO 0x100
0000	040100	GOTO 0x100
0000	000000	NOP
<b>Step 2: Initialize TBLPAG and the read pointer (W6) for TBLRD instruction.</b>		
0000	2007F0	MOV #0x7F, W0
0000	880190	MOV W0, TBLPAG
0000	2xxxx6	MOV #<SourceAddress15:0>, W6
<b>Step 3: Initialize the write pointer (W7) and store the next four locations of code memory to W0:W5.</b>		
0000	EB0380	CLR W7
0000	000000	NOP
0000	BA1BB6	TBLRDL [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BA1BB6	TBLRDL [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BA1BB6	TBLRDL [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BA1BB6	TBLRDL [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
<b>Step 4: Output W0:W5 using the VISI register and REGOUT command.</b>		
0000	883C20	MOV W0, VISI
0000	000000	NOP
0001	<VISI>	Clock out contents of VISI register
0000	000000	NOP
0000	883C21	MOV W1, VISI
0000	000000	NOP
0001	<VISI>	Clock out contents of VISI register
0000	000000	NOP
0000	883C22	MOV W2, VISI
0000	000000	NOP
0001	<VISI>	Clock out contents of VISI register
0000	000000	NOP
0000	883C23	MOV W3, VISI
0000	000000	NOP
0001	<VISI>	Clock out contents of VISI register
0000	000000	NOP
<b>Step 5: Reset device internal PC.</b>		
0000	040100	GOTO 0x100
0000	000000	NOP
<b>Step 6: Repeat steps 3-5 until all desired data memory is read.</b>		



# dsPIC30F Flash Programming Specification

## 11.13 Reading the Application ID Word

The application ID word is stored at address 0x8005BE in executive code memory. To read this memory location, you must use the SIX control code to move this program memory location to the VISI register. The REGOUT control code must then be used to clock the contents of the VISI register out of the device. The corresponding control and instruction codes that must be serially transmitted to the device to perform this operation are shown in [Table 11-13](#).

Once the programmer has clocked-out the application ID word, it must be inspected. If the application ID has the value 0xBB, the programming executive is resident in memory and the device can be programmed using the mechanism described in [Section 5.0 “Device Programming”](#). However, if the application ID has any other value, the programming executive is not resident in memory. It must be loaded to memory before the device can be programmed. The procedure for loading the programming executive to the memory is described in [Section 12.0 “Programming the Programming Executive to Memory”](#).

## 11.14 Exiting ICSP Mode

After confirming that the programming executive is resident in memory, or loading the programming executive, ICSP mode is exited by removing power to the device or bringing MCLR to V<sub>IL</sub>. Programming can then take place by following the procedure outlined in [Section 5.0 “Device Programming”](#).

**TABLE 11-13: SERIAL INSTRUCTION EXECUTION FOR READING THE APPLICATION ID WORD**

Command (Binary)	Data (Hexadecimal)	Description
<b>Step 1: Exit the Reset vector.</b>		
0000	040100	GOTO 0x100
0000	040100	GOTO 0x100
0000	000000	NOP
<b>Step 2: Initialize TBLPAG and the read pointer (W0) for TBLRD instruction.</b>		
0000	200800	MOV #0x80, W0
0000	880190	MOV W0, TBLPAG
0000	205BE0	MOV #0x5BE, W0
0000	207841	MOV VISI, W1
0000	000000	NOP
0000	BA0890	TBLRDL [W0], [W1]
0000	000000	NOP
0000	000000	NOP
<b>Step 3: Output the VISI register using the REGOUT command.</b>		
0001	<VISI>	Clock out contents of the VISI register
0000	000000	NOP

# dsPIC30F Flash Programming Specification

**TABLE 12-1: PROGRAMMING THE PROGRAMMING EXECUTIVE (CONTINUED)**

Command (Binary)	Data (Hexadecimal)	Description
<b>Step 8:</b> Set the read pointer (W6) and load the (next four write) latches.		
0000	EB0300	CLR W6
0000	000000	NOP
0000	BB0BB6	TBLWTL [W6++], [W7]
0000	000000	NOP
0000	000000	NOP
0000	BBDBB6	TBLWTH.B [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BEBBB6	TBLWTH.B [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BB1BB6	TBLWTL [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BB0BB6	TBLWTL [W6++], [W7]
0000	000000	NOP
0000	000000	NOP
0000	BBDBB6	TBLWTH.B [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BEBBB6	TBLWTH.B [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
0000	BB1BB6	TBLWTL [W6++], [W7++]
0000	000000	NOP
0000	000000	NOP
<b>Step 9:</b> Repeat Steps 7-8 eight times to load the write latches for the 32 instructions.		
<b>Step 10:</b> Unlock the NVMCON for programming.		
0000	200558	MOV #0x55, W8
0000	883B38	MOV W8, NVMKEY
0000	200AA9	MOV #0xAA, W9
0000	883B39	MOV W9, NVMKEY
<b>Step 11:</b> Initiate the programming cycle.		
0000	A8E761	BSET NVMCON, #15
0000	000000	NOP
0000	000000	NOP
—	—	Externally time 'P12a' ms (see <a href="#">Section 13.0 “AC/DC Characteristics and Timing Requirements”</a> )
0000	000000	NOP
0000	000000	NOP
0000	A9E761	BCLR NVMCON, #15
0000	000000	NOP
0000	000000	NOP
<b>Step 12:</b> Reset the device internal PC.		
0000	040100	GOTO 0x100
0000	000000	NOP
<b>Step 13:</b> Repeat Steps 7-12 until all 23 rows of executive memory are programmed.		

# dsPIC30F Flash Programming Specification

**TABLE A-1: CHECKSUM COMPUTATION (CONTINUED)**

Device	Read Code Protection	Checksum Computation	Erased Value	Value with 0xAAAAAA at 0x0 and Last Code Address
dsPIC30F5016	Disabled	CFGB+SUM(0:00AFFF)	0xFC06	0xFA08
	Enabled	CFGB	0x0404	0x0404
dsPIC30F6010	Disabled	CFGB+SUM(0:017FFF)	0xC406	0xC208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F6010A	Disabled	CFGB+SUM(0:017FFF)	0xC406	0xC208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F6011	Disabled	CFGB+SUM(0:015FFF)	0xF406	0xF208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F6011A	Disabled	CFGB+SUM(0:015FFF)	0xF406	0xF208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F6012	Disabled	CFGB+SUM(0:017FFF)	0xC406	0xC208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F6012A	Disabled	CFGB+SUM(0:017FFF)	0xC406	0xC208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F6013	Disabled	CFGB+SUM(0:015FFF)	0xF406	0xF208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F6013A	Disabled	CFGB+SUM(0:015FFF)	0xF406	0xF208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F6014	Disabled	CFGB+SUM(0:017FFF)	0xC406	0xC208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F6014A	Disabled	CFGB+SUM(0:017FFF)	0xC406	0xC208
	Enabled	CFGB	0x0404	0x0404
dsPIC30F6015	Disabled	CFGB+SUM(0:017FFF)	0xC406	0xC208
	Enabled	CFGB	0x0404	0x0404

**Item Description:**

**SUM(a:b)** = Byte sum of locations a to b inclusive (all 3 bytes of code memory)

**CFGB** = **Configuration Block (masked)** = Byte sum of ((FOSC&0xC10F) + (FWDT&0x803F) + (FBORPOR&0x87B3) + (FBS&0x310F) + (FSS&0x330F) + (FGS&0x0007) + (FICD&0xC003))

# dsPIC30F Flash Programming Specification

---

## APPENDIX B: HEX FILE FORMAT

Flash programmers process the standard HEX format used by the Microchip development tools. The format supported is the Intel® HEX 32 Format (INHX32). Please refer to Appendix A in the “*MPASM User's Guide*” (DS33014) for more information about hex file formats.

The basic format of the hex file is:

```
:BBAAAATTTHHHH...HHHHCC
```

Each data record begins with a 9-character prefix and always ends with a 2-character checksum. All records begin with ':' regardless of the format. The individual elements are described below.

- **BB** - is a two-digit hexadecimal byte count representing the number of data bytes that appear on the line. Divide this number by two to get the number of words per line.
- **AAAA** - is a four-digit hexadecimal address representing the starting address of the data record. Format is high byte first followed by low byte. The address is doubled because this format only supports 8-bits. Divide the value by two to find the real device address.
- **TT** - is a two-digit record type that will be '00' for data records, '01' for end-of-file records and '04' for extended-address record.
- **HHHH** - is a four-digit hexadecimal data word. Format is low byte followed by high byte. There will be **BB/2** data words following **TT**.
- **CC** - is a two-digit hexadecimal checksum that is the two's complement of the sum of all the preceding bytes in the line record.

Because the Intel hex file format is byte-oriented, and the 16-bit program counter is not, program memory sections require special treatment. Each 24-bit program word is extended to 32 bits by inserting a so-called “phantom byte”. Each program memory address is multiplied by 2 to yield a byte address.

As an example, a section that is located at 0x100 in program memory will be represented in the hex file as 0x200.

The hex file will be produced with the following contents:

```
:020000040000fa
:040200003322110096
:00000001FF
```

Notice that the data record (line 2) has a load address of 0200, while the source code specified address 0x100. Note also that the data is represented in “little-endian” format, meaning the Least Significant Byte (LSB) appears first. The phantom byte appears last, just before the checksum.

## APPENDIX C: REVISION HISTORY

<b>Note:</b> Revision histories were not recorded for revisions A through H. The previous revision (J), was published in August 2007.
---

### Revision K (November 2010)

This version of the document includes the following updates:

- Added Note three to [Section 5.2 “Entering Enhanced ICSP Mode”](#)
- Updated the first paragraph of [Section 10.0 “Device ID”](#)
- Updated [Table 10-1: Device IDs](#)
- Removed the VARIANT bit and updated the bit definition for the DEVID register in [Table 10-2: dsPIC30F Device ID Registers](#)
- Removed the VARIANT bit and updated the bit field definition and description for the DEVID register in [Table 10-3: Device ID Bits Description](#)
- Updated Note 3 in [Section 11.3 “Entering ICSP Mode”](#)
- Updated Step 11 in [Table 11-4: Serial Instruction Execution for Bulk Erasing Program Memory \(Only in Normal-voltage Systems\)](#)
- Updated Steps 5, 12 and 19 in [Table 11-5: Serial Instruction Execution for Erasing Program Memory \(Either in Low-voltage or Normal-voltage Systems\)](#)
- Updated Steps 5, 6 and 8 in [Table 11-7: Serial Instruction Execution for Writing Configuration Registers](#)
- Updated Steps 6 and 8 in [Table 11-8: Serial Instruction Execution for Writing Code Memory](#)
- Updated Steps 6 and 8 in [Table 11-9: Serial Instruction Execution for Writing Data EEPROM](#)
- Updated Entering ICSP™ Mode (see [Figure 11-4](#))
- Updated Steps 4 and 11 in [Table 12-1: Programming the Programming Executive](#)
- Renamed parameters: P12 to P12a and P13 to P13a, and added parameters P12b and P13b in [Table 13-1: AC/DC Characteristics](#)